UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/739,228	12/17/2003	Michael L. Lamb	SJO920030054US1	1120

68431 7590 06/27/2007

TIMOTHY N. ELLIS, PATENT ATTORNEY
8680 VIA MALLORCA
SUITE D
LA JOLLA, CA 92037

EXAMINER	
SAEED, USMAAN	

ART UNIT	PAPER NUMBER
2166	

MAIL DATE	DELIVERY MODE
06/27/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/739,228	LAMB ET AL.	
	Examiner	Art Unit	
	Usmaan Saeed	2166	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *17 April 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *17 December 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 4/17/2007 has been entered.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein

were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

Claims 1-20 are rejected under 35 U.S.C 103(a) as being unpatentable over **Guruprasad Bhat.** (**Bhat** hereinafter) (US PGPub No. 2003/0055808) in view of **Weber et al.** (**Weber** hereinafter) (U.S. PGPub No. 2002/0184360) further in view of **Booth et al.** (**Booth** hereinafter) (U.S. Patent No. 6,493,719).

With respect to claim 1, **Bhat** teaches **a machine readable data storage medium tangibly embodying a program of machine-readable instructions executable by a digital processing apparatus to perform a method for responding to an inquiry, the method comprising the following operations:**

"receiving the inquiry" as log requests may be provided to the logging service by components of the computing system. The logging service may access the property file to determine which storage device incorporated by the computing system is activated as a primary log storage device (**Bhat** Paragraph 0021). Examiner interprets the requests as inquiries.

"obtaining information from a CIMOM" as client API 113 may be an application programming interface used by client application 112 to communicate with CIMOM 142 located in server 140 (**Bhat** Paragraph 0029). A developer uses the CIM specification to describe managed objects and retrieve information about managed objects in server 140 (**Bhat** Paragraph 0030).

"creating at least one Storage Object" as the storage interface processes the request using a proper implementation object based on the type of storage device indicated in the property file and determined by the logging service. The implementation object may be used to perform the detailed functions associated with the actual access of the storage device to complete the logging operation (**Bhat** Paragraph 0021). Examiner interprets the implementation object based on the type of storage device as storage object.

"populating the at least one Storage Object with information received from the CIMOM" as CIMOM 142 communicates with either repository 144 or an appropriate provider 146-1 to 146-N, to obtain information about an object requested by client 140 (**Bhat** Paragraph 0034). This reference is populating an object by obtaining information about an object from CIMOM.

"sending the at least one Storage Object to a calling function" as alternatively, storage interface 210 may be configured to use a loaded implementation object 212-216 to access a storage device 145 and provide information to logging service 141 during, or after, the access (**Bhat** Paragraph 0072). CIMOM 142 may also perform other functions such as setting up communications with repository 144 and

providers 146-1 to 146-N to route requests thereto, security checks, and delivering data from providers 146-1 to 146-N and repository 144 to client 110 (**Bhat** Paragraph 0034).

Bhat teaches the elements of claim 1 as noted above but does not explicitly discloses **"identifying a disk array system as a class of device to be managed," "identifying subcomponents of the disk array system," "receiving a unique ID for the disk array system," "wherein obtaining information from the CIMOM includes, given the unique ID for the disk array system, obtaining information regarding all component storage pools of the disk array system, and obtaining information regarding all component volumes of the disk array system," "wherein creating the at least one storage object includes identifying entities attached to the disk array system, and identifying parent-child relationships between the entities," "wherein the at least one storage object includes a storage object corresponding with the disk array system" and "wherein obtaining information from the CIMOM further includes obtaining, in one step: information about all disk array systems managed by the CIMOM, and information about all volumes, disks, disk groups, and storage pools corresponding respectively with each of the disk array systems managed by the CIMOM, and information about relationships between all of the corresponding volumes, disks, disk groups, and storage pools."**

However, **Weber** discloses **"identifying a disk array system as a class of device to be managed"** as (**Weber** Paragraph 0032), **"identifying subcomponents of the disk array system"** as (**Weber** Paragraph 0033), **"receiving a unique ID for the disk array system"** as (**Weber** Figure 2 & 3), **"wherein obtaining information**

from the CIMOM includes, given the unique ID for the disk array system, obtaining information regarding all component storage pools of the disk array system, and obtaining information regarding all component volumes of the disk array system" as (**Weber** Paragraph 0103), **"wherein creating the at least one storage object includes identifying entities attached to the disk array system, and identifying parent-child relationships between the entities"** as (**Weber** Paragraph 0091), **and "wherein the at least one storage object includes a storage object corresponding with the disk array system"** as (**Weber** Figures 4 & 5).

"information about all disk array systems managed by the CIMOM, and information about all volumes, disks, disk groups, and storage pools corresponding respectively with each of the disk array systems managed by the CIMOM, and information about relationships between all of the corresponding volumes, disks, disk groups, and storage pools" as (**Weber** Paragraph 0103, 0032, 0108).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because **Weber's** teaching would have allowed **Bhat** to express the requests from management interface in terms of device object model, which interprets the requests and carries out the requests by interacting with RAID engine 530 and then respond back to the management interface applet 518 in terms of the object model (**Weber** Paragraph 0071).

Bhat and Weber teach the elements of claims 1 as noted above but do not explicitly disclose, **"wherein obtaining information from the CIMOM further includes obtaining, in one step."**

However, **Booth** teaches **"wherein obtaining information from the CIMOM further includes obtaining, in one step"** as the CIMOM then communicates with one or more sources of the information, known as providers, to return an appropriate reply. The CIMOM is intelligent in that it can decompose queries into requests from multiple providers and synthesize the results into a single response, filter excess information, work with the capabilities of the providers, and so forth (**Booth** Col 1, Lines 42-49). Examiner interprets the single response as obtaining information in one step.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because **Booth's** teaching would have allowed **Bhat and Weber** to provide scripting which enables a set of objects or properties to be serviced iteratively, for example to manipulate or retrieve properties for a set of resources in a simple loop.

Claims 19 and 20 are essentially the same as claim 1 except they set forth the claimed invention as a system and a method and are rejected for the same reasons as applied hereinabove.

With respect to claim 2, **Bhat** teaches **"the machine readable data storage medium of claim 1, wherein the obtaining operation comprises using a CIM Client API to obtain requested information from the CIMOM"** as client API 113 may be an application programming interface used by client application 112 to communicate with CIMOM 142 located in server 140 (**Bhat** Paragraph 0029). A developer uses the CIM specification to describe managed objects and retrieve information about managed objects in server 140 (**Bhat** Paragraph 0030).

With respect to claim 3, **Bhat** teaches **"the machine readable data storage medium of claim 1, wherein the operation of creating at least one Storage Object comprises creating a set of Storage Objects"** as a logging service may be configured to interact with a storage interface that uses implementation objects that are each associated with a particular type of storage device incorporated within the computing system. Each implementation object may be configured to use processes specific to a particular type of storage device and may be used by the logging service to access the storage device (**Bhat** Paragraph 0011).

With respect to claim 4, **Bhat** teaches **a machine readable data storage medium tangibly embodying a program of machine-readable instructions executable by a digital processing apparatus to perform a method for responding to an inquiry, the method comprising the following operations:**

"receiving the inquiry" as log requests may be provided to the logging service by components of the computing system. The logging service may access the property file to determine which storage device incorporated by the computing system is activated as a primary log storage device (**Bhat** Paragraph 0021). Examiner interprets the requests as inquiries.

"obtaining information from a CIMOM" as client API 113 may be an application programming interface used by client application 112 to communicate with CIMOM 142 located in server 140 (**Bhat** Paragraph 0029). A developer uses the CIM specification to describe managed objects and retrieve information about managed objects in server 140 (**Bhat** Paragraph 0030).

"creating at least one Storage Object" as the storage interface processes the request using a proper implementation object based on the type of storage device indicated in the property file and determined by the logging service. The implementation object may be used to perform the detailed functions associated with the actual access of the storage device to complete the logging operation (**Bhat** Paragraph 0021). Examiner interprets the implementation object based on the type of storage device as storage object.

"populating the at least one Storage Object with information received from the CIMOM" as CIMOM 142 communicates with either repository 144 or an appropriate provider 146-1 to 146-N, to obtain information about an object requested by client 140 (**Bhat** Paragraph 0034). This reference is populating an object by obtaining information about an object from CIMOM.

"sending the at least one Storage Object to a calling function" as alternatively, storage interface 210 may be configured to use a loaded implementation object 212-216 to access a storage device 145 and provide information to logging service 141 during, or after, the access (**Bhat** Paragraph 0072). CIMOM 142 may also perform other functions such as setting up communications with repository 144 and providers 146-1 to 146-N to route requests thereto, security checks, and delivering data from providers 146-1 to 146-N and repository 144 to client 110 (**Bhat** Paragraph 0034).

"wherein the operations are performed as an intermediary between a CIM Client application and a CIM API" as client API 113 may be an application programming interface used by client application 112 to communicate with CIMOM 142 located in server 140 (**Bhat** Paragraph 0029, 0028). A developer uses the CIM specification to describe managed objects and retrieve information about managed objects in server 140 (**Bhat** Paragraph 0030).

"wherein each Storage Object is created by using a Java package comprising classes that define a plurality of storage entity objects" as client API 113 may represent and manipulate CIM objects. These objects may be represented in software written in an object-oriented programming language, such as the Java.TM. programming language. An object may be a computer representation or model of a managed resource of server 140, such as a printer, disk drive, and CPU. A developer uses the CIM specification to describe managed objects and retrieve information about managed objects in server 140 (**Bhat** Paragraph 0030 & Paragraph 0036).

"wherein the plurality of storage entity objects include Disk Array System, Storage Pool, Volume, Host System, FCPort, and Disk, objects" as the term "memory" used with memory implementation object 212 and memory storage device 230 may be associated with semiconductor type memories, such as RAM, ROM, SRAM, DRAM, DRAM, EPROM, NVRAM, or the like. The term "file" used in conjunction with file implementation object 214 and file storage device 240 may be associated with magnetic disk devices. And, the term "tape" used in conjunction with tape implementation object 216 and tape storage device 250 may be associated with magnetic tape storage devices. It should be noted, however, that the above examples are not intended to be limiting and any number of various types of storage devices, such as optical disks, (and their associated implementation objects) may be implemented by systems and methods consistent with features of the present invention, without departing from the scope of the invention.

"wherein the creating operation comprises creating a plurality of Storage Objects" as client API 113 may represent and manipulate CIM objects. These objects may be represented in software written in an object-oriented programming language, such as the Java.TM. programming language. An object may be a computer representation or model of a managed resource of server 140, such as a printer, disk drive, and CPU. A developer uses the CIM specification to describe managed objects and retrieve information about managed objects in server 140 (**Bhat** Paragraph 0030) **"and wherein properties of each Storage Object map directly to properties of at least one CIM Class used to represent a corresponding storage entity"** as

providers 146-1 to 146-N may be classes that perform various functions in response to a request from CIMOM 142 and act as intermediaries between CIMOM 142 and one or more managed devices. For instance, providers 146-1 to 146-N may map information from a managed device to a CIM class that may be written in an object oriented language, such as the Java programming language (**Bhat** Paragraph 0036).

"wherein the inquiry is received from a SRM CIM Client Application" as server 140 may execute software applications and processes that perform tasks similar to that of client 110. Accordingly, these applications and processes may provide requests to CIMOM 142 associated with a managed resource as well. Furthermore, methods, systems and articles of manufacture consistent with features of the present invention are not limited to CIMOM 142 receiving requests from client 110 alone. Requests from other sources, such as components within server 140 and entities outside of server 140 may be processed by CIMOM 142 (**Bhat** Paragraph 0044).

"a request for all storage entities of a specified type associated with the designated storage entity" as the storage interface processes the request using a proper implementation object based on the type of storage device indicated in the property file and determined by the logging service. The implementation object may be used to perform the detailed functions associated with the actual access of the storage device to complete the logging operation (**Bhat** Paragraph 0021).

"receiving, obtaining, creating, populating, and sending to obtain information concerning the identified storage entity" as client API 113 may be an application programming interface used by client application 112 to communicate with

CIMOM 142 located in server 140 (**Bhat** Paragraph 0029). A developer uses the CIM specification to describe managed objects and retrieve information about managed objects in server 140 (**Bhat** Paragraph 0030).

Bhat teaches the elements of claim 4 as noted above but does not explicitly discloses **"identifying a disk array system as a class of device to be managed," "identifying subcomponents of the disk array system," "receiving a unique ID for the disk array system," "wherein obtaining information from the CIMOM includes, given the unique ID for the disk array system, obtaining information regarding all component storage pools of the disk array system, and obtaining information regarding all component volumes of the disk array system," "wherein creating the at least one storage object includes identifying entities attached to the disk array system, and identifying parent-child relationships between the entities," "wherein the at least one storage object includes a storage object corresponding with the disk array system," "wherein obtaining information from the CIMOM further includes obtaining, in one step: information about all disk array systems managed by the CIMOM, and information about all volumes, disks, disk groups, and storage pools corresponding respectively with each of the disk array systems managed by the CIMOM, and information about relationships between all of the corresponding volumes, disks, disk groups, and storage pools," "plurality of storage entity objects include Disk Array System, Storage Pool, Volume, Host System, FCPort, and Disk, objects," "wherein the Disk Array System object is a top level object, and wherein at least one object other than the Disk Array System**

object is a subcomponent of an object other than the Disk Array System object," "wherein the creating operation comprises creating a plurality of Storage Objects, and wherein the Storage Objects have associations to each other that are consistent with corresponding storage entities' relationships modeled in a SMI/Bluefin profile," "wherein receiving the inquiry includes a unique ID for storage pool and the operations further comprise obtaining a storage object corresponding with the storage pool, given the unique ID for the storage pool," "wherein the inquiry includes the unique ID of a designated storage entity," "wherein the inquiry includes the unique ID of an identified top level storage entity and wherein the receiving, obtaining, creating, populating, and sending operations are repeated to obtain information concerning the identified top level storage entity and all of the components of the identified top level storage entity,"

However, **Weber** discloses **"identifying a disk array system as a class of device to be managed"** as (**Weber** Paragraph 0032), **"identifying subcomponents of the disk array system"** as (**Weber** Paragraph 0033), **"receiving a unique ID for the disk array system"** as (**Weber** Figure 2 & 3), **"wherein obtaining information from the CIMOM includes, given the unique ID for the disk array system, obtaining information regarding all component storage pools of the disk array system, and obtaining information regarding all component volumes of the disk array system"** as (**Weber** Paragraph 0103), **"wherein creating the at least one storage object includes identifying entities attached to the disk array system, and identifying parent-child relationships between the entities"** as (**Weber** Paragraph

0091), **and "wherein the at least one storage object includes a storage object corresponding with the disk array system"** as (**Weber** Figures 4 & 5).

"information about all disk array systems managed by the CIMOM, and information about all volumes, disks, disk groups, and storage pools corresponding respectively with each of the disk array systems managed by the CIMOM, and information about relationships between all of the corresponding volumes, disks, disk groups, and storage pools" as (**Weber** Paragraph 0103, 0032, 0108).

"plurality of storage entity objects include Disk Array System, Storage Pool, Volume, Host System, FCPort, and Disk, objects" as aspects of an array device that may be updated include individual object revision definitions for drive groups, drives, volumes, redundant controllers, storage systems, and the like (**Weber** Paragraph 0044, Figure 1 & 7).

"wherein the Disk Array System object is a top level object, and wherein at least one object other than the Disk Array System object is a subcomponent of an object other than the Disk Array System object" as the logical composition and properties of the selected device (e.g., storage array). The logical objects of the storage array are organized into a tree structure to make their interrelationships apparent. Screen 700 illustrates an example of a typical set of logical objects, including volume groups 706, volumes 708, free capacity regions 710, and unassigned capacity 712 (**Weber** Paragraph 0091). Aspects of an array device that may be updated include

individual object revision definitions for drive groups, drives, volumes, redundant controllers, storage systems, and the like (**Weber** Paragraph 0044, Figure 1 & 7).

"wherein the creating operation comprises creating a plurality of Storage Objects, and wherein the Storage Objects have associations to each other that are consistent with corresponding storage entities' relationships modeled in a SMI/Bluefin profile" as the logical composition and properties of the selected device (e.g., storage array). The logical objects of the storage array are organized into a tree structure to make their interrelationships apparent. Screen 700 illustrates an example of a typical set of logical objects, including volume groups 706, volumes 708, free capacity regions 710, and unassigned capacity 712 (**Weber** Paragraph 0091). Aspects of an array device that may be updated include individual object revision definitions for drive groups, drives, volumes, redundant controllers, storage systems, and the like (**Weber** Paragraph 0044, Figure 1 & 7).

"wherein receiving the inquiry includes a unique ID for storage pool and the operations further comprise obtaining a storage object corresponding with the storage pool, given the unique ID for the storage pool" as (**Weber** Figures 2 &3, Paragraph 0103).

"wherein the inquiry includes the unique ID of a designated storage entity" as Figures 2 & 3, reference numerals 204-1 and 204-2 (**Weber** Figures 2 &3).

"wherein the inquiry includes the unique ID of an identified top level storage entity" as Figures 2 & 3, reference numerals 204-1 and 204-2 (**Weber** Figures 2 &3) **"to obtain information concerning the identified top level storage entity and**

all of the components of the identified top level storage entity" as the logical composition and properties of the selected device (e.g., storage array). The logical objects of the storage array are organized into a tree structure to make their interrelationships apparent. Screen 700 illustrates an example of a typical set of logical objects, including volume groups 706, volumes 708, free capacity regions 710, and unassigned capacity 712 (**Weber** Paragraph 0091). Aspects of an array device that may be updated include individual object revision definitions for drive groups, drives, volumes, redundant controllers, storage systems, and the like (**Weber** Paragraph 0044, Figure 1 & 7).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because **Weber's** teaching would have allowed **Bhat** to express the requests from management interface in terms of device object model, which interprets the requests and carries out the requests by interacting with RAID engine 530 and then respond back to the management interface applet 518 in terms of the object model (**Weber** Paragraph 0071).

Bhat and Weber teach the elements of claims 4 as noted above but do not explicitly disclose, **"wherein obtaining information from the CIMOM further includes obtaining, in one step," "operations are repeated to obtain information concerning the identified storage entity."**

However, **Booth** teaches **"wherein obtaining information from the CIMOM further includes obtaining, in one step"** as the CIMOM then communicates with one

or more sources of the information, known as providers, to return an appropriate reply. The CIMOM is intelligent in that it can decompose queries into requests from multiple providers and synthesize the results into a single response, filter excess information, work with the capabilities of the providers, and so forth (**Booth** Col 1, Lines 42-49). Examiner interprets the single response as obtaining information in one step.

"wherein obtaining information from the CIMOM further includes obtaining, in one step" as the CIMOM then communicates with one or more sources of the information, known as providers, to return an appropriate reply. The CIMOM is intelligent in that it can decompose queries into requests from multiple providers and synthesize the results into a single response, filter excess information, work with the capabilities of the providers, and so forth (**Booth** Col 1, Lines 42-49). Examiner interprets the single response as obtaining information in one step.

Further, **Booth** teaches, **"wherein the operations are performed as an intermediary between a CIM Client application and a CIM API"** as (**Booth** Figures 2 and 3).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because **Booth's** teaching would have allowed **Bhat and Weber** to provide scripting which enables a set of objects or properties to be serviced iteratively, for example to manipulate or retrieve properties for a set of resources in a simple loop.

With respect to claim 5, **Bhat** teaches **"the machine readable data storage medium of claim 1, wherein each Storage Object is created by using a Java package comprising classes that define a plurality of storage entity objects"** as client API 113 may represent and manipulate CIM objects. These objects may be represented in software written in an object-oriented programming language, such as the Java.TM. programming language. An object may be a computer representation or model of a managed resource of server 140, such as a printer, disk drive, and CPU. A developer uses the CIM specification to describe managed objects and retrieve information about managed objects in server 140 (**Bhat** Paragraph 0030 & Paragraph 0036).

With respect to claim 6, **Bhat** teaches **"the machine readable data storage medium of claim 5, wherein the plurality of storage entity objects include Disk Array System, Storage Pool, Volume, Host System, FCPort, and Disk, objects"** as the term "memory" used with memory implementation object 212 and memory storage device 230 may be associated with semiconductor type memories, such as RAM, ROM, SRAM, DRAM, DRAM, EPROM, NVRAM, or the like. The term "file" used in conjunction with file implementation object 214 and file storage device 240 may be associated with magnetic disk devices. And, the term "tape" used in conjunction with tape implementation object 216 and tape storage device 250 may be associated with magnetic tape storage devices. It should be noted, however, that the above examples are not intended to be limiting and any number of various types of storage devices, such

as optical disks, (and their associated implementation objects) may be implemented by systems and methods consistent with features of the present invention, without departing from the scope of the invention.

Bhat teaches elements of claim 6 as noted above but does not explicitly disclose **"plurality of storage entity objects include Disk Array System, Storage Pool, Volume, Host System, FCPort, and Disk, objects."**

However, **Weber** discloses **"plurality of storage entity objects include Disk Array System, Storage Pool, Volume, Host System, FCPort, and Disk, objects"** as aspects of an array device that may be updated include individual object revision definitions for drive groups, drives, volumes, redundant controllers, storage systems, and the like (**Weber** Paragraph 0044, Figure 1 & 7).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because **Weber's** teaching would have allowed **Bhat** to express the requests from management interface in terms of device object model, which interprets the requests and carries out the requests by interacting with RAID engine 530 and then respond back to the management interface applet 518 in terms of the object model (**Weber** Paragraph 0071).

With respect to claim 7, **Bhat** does not explicitly disclose **"the machine readable data storage medium of claim 6, wherein the Disk Array System object is a top**

level object, and wherein each object other than the Disk Array System object is associated as a component of the Disk Array System object."

However, **Weber** discloses **"the machine readable data storage medium of claim 6, wherein the Disk Array System object is a top level object, and wherein each object other than the Disk Array System object is associated as a component of the Disk Array System object"** as the logical composition and properties of the selected device (e.g., storage array). The logical objects of the storage array are organized into a tree structure to make their interrelationships apparent. Screen 700 illustrates an example of a typical set of logical objects, including volume groups 706, volumes 708, free capacity regions 710, and unassigned capacity 712 (**Weber** Paragraph 0091). Aspects of an array device that may be updated include individual object revision definitions for drive groups, drives, volumes, redundant controllers, storage systems, and the like (**Weber** Paragraph 0044, Figure 1 & 7).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because **Weber's** teaching would have allowed **Bhat** to express the requests from management interface in terms of device object model, which interprets the requests and carries out the requests by interacting with RAID engine 530 and then respond back to the management interface applet 518 in terms of the object model (**Weber** Paragraph 0071).

With respect to claim 8, **Bhat** does not explicitly disclose **"the machine readable data storage medium of claim 6, wherein the Disk Array System object is a top level object, and wherein at least one object other than the Disk Array System object is a subcomponent of an object other than the Disk Array System object."**

However, **Weber** discloses **"the machine readable data storage medium of claim 6, wherein the Disk Array System object is a top level object, and wherein at least one object other than the Disk Array System object is a subcomponent of an object other than the Disk Array System object"** as the logical composition and properties of the selected device (e.g., storage array). The logical objects of the storage array are organized into a tree structure to make their interrelationships apparent. Screen 700 illustrates an example of a typical set of logical objects, including volume groups 706, volumes 708, free capacity regions 710, and unassigned capacity 712 (**Weber** Paragraph 0091). Aspects of an array device that may be updated include individual object revision definitions for drive groups, drives, volumes, redundant controllers, storage systems, and the like (**Weber** Paragraph 0044, Figure 1 & 7).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because **Weber's** teaching would have allowed **Bhat** to express the requests from management interface in terms of device object model, which interprets the requests and carries out the requests by interacting with RAID engine 530 and then respond back to the management interface applet 518 in terms of the object model (**Weber** Paragraph 0071).

With respect to claim 9, **Bhat** does not explicitly disclose, **"the machine readable data storage medium of claim 1, wherein the creating operation comprises creating a plurality of Storage Objects, and wherein the Storage Objects have associations to each other that are consistent with corresponding storage entities' relationships modeled in a SMI/Bluefin profile."**

However, **Weber** discloses **"the machine readable data storage medium of claim 1, wherein the creating operation comprises creating a plurality of Storage Objects, and wherein the Storage Objects have associations to each other that are consistent with corresponding storage entities' relationships modeled in a SMI/Bluefin profile"** as the logical composition and properties of the selected device (e.g., storage array). The logical objects of the storage array are organized into a tree structure to make their interrelationships apparent. Screen 700 illustrates an example of a typical set of logical objects, including volume groups 706, volumes 708, free capacity regions 710, and unassigned capacity 712 (**Weber** Paragraph 0091). Aspects of an array device that may be updated include individual object revision definitions for drive groups, drives, volumes, redundant controllers, storage systems, and the like (**Weber** Paragraph 0044, Figure 1 & 7).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because **Weber's** teaching would have allowed **Bhat** to express the requests from management interface in terms of device object model, which interprets the requests and carries out the

requests by interacting with RAID engine 530 and then respond back to the management interface applet 518 in terms of the object model (**Weber** Paragraph 0071).

With respect to claim 10, **Bhat** teaches **"the machine readable data storage medium of claim 1, wherein the creating operation comprises creating a plurality of Storage Objects"** as client API 113 may represent and manipulate CIM objects. These objects may be represented in software written in an object-oriented programming language, such as the Java.TM. programming language. An object may be a computer representation or model of a managed resource of server 140, such as a printer, disk drive, and CPU. A developer uses the CIM specification to describe managed objects and retrieve information about managed objects in server 140 (**Bhat** Paragraph 0030) **"and wherein properties of each Storage Object map directly to properties of at least one CIM Class used to represent a corresponding storage entity"** as providers 146-1 to 146-N may be classes that perform various functions in response to a request from CIMOM 142 and act as intermediaries between CIMOM 142 and one or more managed devices. For instance, providers 146-1 to 146-N may map information from a managed device to a CIM class that may be written in an object oriented language, such as the Java programming language (**Bhat** Paragraph 0036).

With respect to claim 11, **Bhat** teaches **"the machine readable data storage medium of claim 1, wherein the inquiry is received from a SRM CIM Client**

Application" as server 140 may execute software applications and processes that perform tasks similar to that of client 110. Accordingly, these applications and processes may provide requests to CIMOM 142 associated with a managed resource as well. Furthermore, methods, systems and articles of manufacture consistent with features of the present invention are not limited to CIMOM 142 receiving requests from client 110 alone. Requests from other sources, such as components within server 140 and entities outside of server 140 may be processed by CIMOM 142 (**Bhat** Paragraph 0044).

With respect to claim 12, **Bhat** teaches **"the machine readable data storage medium of claim 1, wherein the inquiry is received from a CIM Discover Tool"** as requests from other sources, such as components within server 140 and entities outside of server 140 may be processed by CIMOM 142 (**Bhat** Paragraph 0044). Alternatively, the requests may originate from sources other than client 110, such as an application or process executed within server 140 (**Bhat** Paragraph 0051).

With respect to claim 13, **Bhat** does not explicitly teaches, **"the machine readable data storage medium of claim 1, wherein receiving the inquiry includes a unique ID for storage pool and the operations further comprise obtaining a storage object corresponding with the storage pool, given the unique ID for the storage pool."**

However, **Weber** discloses, **"wherein receiving the inquiry includes a unique ID for storage pool and the operations further comprise obtaining a storage object corresponding with the storage pool, given the unique ID for the storage pool"** as (**Weber** Figures 2 &3, Paragraph 0103).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because **Weber's** teaching would have allowed **Bhat** to express the requests from management interface in terms of device object model, which interprets the requests and carries out the requests by interacting with RAID engine 530 and then respond back to the management interface applet 518 in terms of the object model (**Weber** Paragraph 0071).

With respect to claim 14, **Bhat** teaches **"and is a request for all storage entities of a specified type associated with the designated storage entity"** as the storage interface processes the request using a proper implementation object based on the type of storage device indicated in the property file and determined by the logging service. The implementation object may be used to perform the detailed functions associated with the actual access of the storage device to complete the logging operation (**Bhat** Paragraph 0021).

Bhat teaches the elements of claim 14 as noted above but does not explicitly disclose the step of **"wherein the inquiry includes the unique ID of a designated storage entity."**

However, **Weber** discloses, **"wherein the inquiry includes the unique ID of a designated storage entity"** as Figures 2 & 3, reference numerals 204-1 and 204-2 (**Weber** Figures 2 &3).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because **Weber's** teaching would have allowed **Bhat** to express the requests from management interface in terms of device object model, which interprets the requests and carries out the requests by interacting with RAID engine 530 and then respond back to the management interface applet 518 in terms of the object model (**Weber** Paragraph 0071).

With respect to claim 15, **Bhat** teaches **"information identifying a specific CIMOM"** as CIMOM 142, and its functionalities, such as logging service 141, may be provided by a vendor (not shown) over network 120 to server 140. Server 140 may download or retrieve CIMOM 142 from the vendor using well known network data transfer means (Bhat Paragraph 0046) **"and storage entity type that are managed by the identified CIMOM"** as a CIM Object Manager (CIMOM) located at a remote server. A CIMOM is a process responsible for handling all CIM related communications between a client and the server where the CIMOM is located (**Bhat** Paragraph 0008). The storage interface processes the request using a proper implementation object based on the type of storage device indicated in the property file and determined by the logging service. The implementation object may be used to perform the detailed

functions associated with the actual access of the storage device to complete the logging operation (**Bhat** Paragraph 0021).

Bhat teaches the elements of claim 15 as noted above but does not explicitly disclose the step of **"the machine readable data storage medium of claim 1, wherein the inquiry includes information identifying a top level storage entity type and information identifying a specific CIMOM, and is a request for information about all entities of the identified top level storage entity type that are managed by the identified CIMOM."**

However, **Weber** discloses **"the machine readable data storage medium of claim 1, wherein the inquiry includes information identifying a top level storage entity type and information identifying a specific CIMOM, and is a request for information about all entities of the identified top level storage entity type that are managed by the identified CIMOM"** as the logical composition and properties of the selected device (e.g., storage array). The logical objects of the storage array are organized into a tree structure to make their interrelationships apparent. Screen 700 illustrates an example of a typical set of logical objects, including volume groups 706, volumes 708, free capacity regions 710, and unassigned capacity 712 (**Weber** Paragraph 0091). Aspects of an array device that may be updated include individual object revision definitions for drive groups, drives, volumes, redundant controllers, storage systems, and the like (**Weber** Paragraph 0044, Figure 1 & 7).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because **Weber's**

teaching would have allowed **Bhat** to express the requests from management interface in terms of device object model, which interprets the requests and carries out the requests by interacting with RAID engine 530 and then respond back to the management interface applet 518 in terms of the object model (**Weber** Paragraph 0071).

With respect to claim 16, **Bhat** teaches **"receiving, obtaining, creating, populating, and sending to obtain information concerning the identified storage entity"** as client API 113 may be an application programming interface used by client application 112 to communicate with CIMOM 142 located in server 140 (**Bhat** Paragraph 0029). A developer uses the CIM specification to describe managed objects and retrieve information about managed objects in server 140 (**Bhat** Paragraph 0030).

Bhat teaches the elements of claim 16 as noted above but does not explicitly disclose the **"wherein the inquiry includes the unique ID of an identified top level storage entity and wherein the receiving, obtaining, creating, populating, and sending operations are repeated to obtain information concerning the identified top level storage entity and all of the components of the identified top level storage entity."**

However, **Weber** discloses **"wherein the inquiry includes the unique ID of an identified top level storage entity"** as Figures 2 & 3, reference numerals 204-1 and 204-2 (**Weber** Figures 2 &3) **"to obtain information concerning the identified top level storage entity and all of the components of the identified top level storage**

entity" as the logical composition and properties of the selected device (e.g., storage array). The logical objects of the storage array are organized into a tree structure to make their interrelationships apparent. Screen 700 illustrates an example of a typical set of logical objects, including volume groups 706, volumes 708, free capacity regions 710, and unassigned capacity 712 (**Weber** Paragraph 0091). Aspects of an array device that may be updated include individual object revision definitions for drive groups, drives, volumes, redundant controllers, storage systems, and the like (**Weber** Paragraph 0044, Figure 1 & 7).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because **Weber's** teaching would have allowed **Bhat** to express the requests from management interface in terms of device object model, which interprets the requests and carries out the requests by interacting with RAID engine 530 and then respond back to the management interface applet 518 in terms of the object model (**Weber** Paragraph 0071).

Bhat and Weber teach the elements of claim 16 as noted above but do not explicitly disclose the step of **"operations are repeated to obtain information concerning the identified storage entity."**

However, **Booth** discloses **"operations are repeated to obtain information concerning the identified storage entity"** as collections enable a set of objects to be serviced iteratively, for example, to manipulate or retrieve properties for a set of resources in simple loop (**Booth** Abstract).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because **Booth's** teaching would have allowed **Bhat and Weber** to provide scripting which enables a set of objects or properties to be serviced iteratively, for example to manipulate or retrieve properties for a set of resources in a simple loop and to synthesize results into a single response.

With respect to claim 17, **Bhat** teaches **"receiving, obtaining, creating, populating, and sending to obtain information concerning the component storage entity"** as client API 113 may be an application programming interface used by client application 112 to communicate with CIMOM 142 located in server 140 (**Bhat** Paragraph 0029). A developer uses the CIM specification to describe managed objects and retrieve information about managed objects in server 140 (**Bhat** Paragraph 0030).

Bhat teaches the elements of claim 17 as noted above but does not explicitly disclose the **"the machine readable data storage medium of claim 1, wherein the inquiry includes the unique ID of a component storage entity, and wherein the receiving, obtaining, creating, populating, and sending operations are repeated to obtain information concerning the component storage entity and subcomponents of the component storage entity."**

However, **Weber** discloses **"the machine readable data storage medium of claim 1, wherein the inquiry includes the unique ID of a component storage entity"** as Figures 2 & 3, reference numerals 204-1 and 204-2 (**Weber** Figures 2 &3)

"and wherein the receiving, obtaining, creating, populating, and sending operations are repeated to obtain information concerning the component storage entity and subcomponents of the component storage entity." as the logical composition and properties of the selected device (e.g., storage array). The logical objects of the storage array are organized into a tree structure to make their interrelationships apparent. Screen 700 illustrates an example of a typical set of logical objects, including volume groups 706, volumes 708, free capacity regions 710, and unassigned capacity 712 (**Weber** Paragraph 0091). Aspects of an array device that may be updated include individual object revision definitions for drive groups, drives, volumes, redundant controllers, storage systems, and the like (**Weber** Paragraph 0044, Figure 1 & 7).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because **Weber's** teaching would have allowed **Bhat** to express the requests from management interface in terms of device object model, which interprets the requests and carries out the requests by interacting with RAID engine 530 and then respond back to the management interface applet 518 in terms of the object model (**Weber** Paragraph 0071).

Bhat and Weber teach the elements of claim 17 as noted above but do not explicitly disclose the step of **"operations are repeated to obtain information concerning the component storage entity."**

However, **Booth** discloses **"operations are repeated to obtain information concerning the component storage entity"** as collections enable a set of objects to be serviced iteratively, for example, to manipulate or retrieve properties for a set of resources in simple loop (**Booth** Abstract).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because **Booth's** teaching would have allowed **Bhat and Weber** to provide scripting which enables a set of objects or properties to be serviced iteratively, for example to manipulate or retrieve properties for a set of resources in a simple loop and to synthesize results into a single response.

With respect to claim 18, **Bhat** discloses **"receiving, obtaining, creating, populating, and sending to obtain information concerning the component storage entity"** as client API 113 may be an application programming interface used by client application 112 to communicate with CIMOM 142 located in server 140 (**Bhat** Paragraph 0029). A developer uses the CIM specification to describe managed objects and retrieve information about managed objects in server 140 (**Bhat** Paragraph 0030).

Bhat teaches the elements of claim 18 as noted above but does not explicitly disclose the **"the machine readable data storage medium of claim 1, wherein the inquiry includes the unique ID of a component storage entity, and wherein the receiving, obtaining, creating, populating, and sending operations are repeated to**

obtain information concerning the component storage entity and the component storage entity's relationships to other components."

However, **Weber** discloses **"the machine readable data storage medium of claim 1, wherein the inquiry includes the unique ID of a component storage entity"** as Figures 2 & 3, reference numerals 204-1 and 204-2 (**Weber** Figures 2 &3) **"and wherein the receiving, obtaining, creating, populating, and sending operations are repeated to obtain information concerning the component storage entity and the component storage entity's relationships to other components"** as the logical composition and properties of the selected device (e.g., storage array). The logical objects of the storage array are organized into a tree structure to make their interrelationships apparent. Screen 700 illustrates an example of a typical set of logical objects, including volume groups 706, volumes 708, free capacity regions 710, and unassigned capacity 712 (**Weber** Paragraph 0091). Aspects of an array device that may be updated include individual object revision definitions for drive groups, drives, volumes, redundant controllers, storage systems, and the like (**Weber** Paragraph 0044, Figure 1 & 7).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because **Weber's** teaching would have allowed **Bhat** to express the requests from management interface in terms of device object model, which interprets the requests and carries out the requests by interacting with RAID engine 530 and then respond back to the

management interface applet 518 in terms of the object model (**Weber** Paragraph 0071).

Bhat and Weber teach the elements of claim 18 as noted above but do not explicitly disclose the step of **"operations are repeated to obtain information concerning the component storage entity."**

However, **Booth** discloses **"operations are repeated to obtain information concerning the component storage entity"** as collections enable a set of objects to be serviced iteratively, for example, to manipulate or retrieve properties for a set of resources in simple loop (**Booth** Abstract).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because **Booth's** teaching would have allowed **Bhat and Weber** to provide scripting which enables a set of objects or properties to be serviced iteratively, for example to manipulate or retrieve properties for a set of resources in a simple loop and to synthesize results into a single response.

Response to Arguments

Applicant's arguments filed on 4/17/2007 have been considered but are moot in view of the new ground(s) of rejection.

See above rejections for the arguments. In these arguments applicant relies on the amended claims and not the original ones.

Claims must be given the broadest reasonable interpretation during examination and limitations appearing in the specification but not recited in the claim are not read into the claim (See M.P.E.P. 2111 [R-I]).

Contact Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Usmaan Saeed whose telephone number is (571)272-4046. The examiner can normally be reached on M-F 8-5.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hosain Alam can be reached on (571)272-3978. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Usmaan Saeed
Patent Examiner
Art Unit: 2166

Leslie Wong LW
Primary Examiner

US
June 18, 2007

MOHAMMAD ALI
PRIMARY EXAMINER